

Design and Implementation of a Magic Mirror Using Kinect



Thesis Submitted in partial fulfillment of the requirement for the degree of

Bachelor of Computer Science and Engineering

Under the Supervision of

Dr. Jia Uddin

By

Md. Moniruzzaman Monir 11201031

Md. Nafis Hasan Siddique 11201002

Afsana Pervin Tanni 11201033

Md. Nahyan Ebn Hashem 15101143

School of Engineering and Computer Science

BRAC University

Submitted on: August, 2016

DECLARATION

We, hereby declare that this thesis is based on the results found by ourselves. Materials of work found by other researcher are mentioned by reference. This Thesis, neither in whole or in part, has been previously submitted for any degree.

Signature of Supervisor

Signature of Author

Dr. Jia Uddin

Assistant Professor

Department of Computer Science and

Engineering,

BRAC University

Md. Moniruzzaman Monir

Md. Nahyan Ebn Hashem

Md. Nafis Hasan Siddique

Afsana Pervin Tanni

ACKNOWLEDGEMENTS

All thanks to Almighty ALLAH, the creator and the owner of this universe, the most merciful, beneficent and the most gracious who provided us guidance, strength and abilities to complete this research.

We are especially thankful to Dr. Jia Uddin, our thesis supervisor, for his help, guidance and support in completion of our project. We also thankful to the BRAC University Faculty Staffs of the Computer Science and Engineering, who have been a light of guidance for us in the whole study period at BRAC University, particularly in building our base in education and enhancing our knowledge.

Finally, we would like to express our sincere gratefulness to our beloved parents, brothers and sisters for their love and care. We are grateful to all of our friends who helped us directly or indirectly to complete our thesis.

CONTENTS

DECLARATION	ii
ACKNOWLEDGEMENTS.....	iii
CONTENTS	iv
LIST OF FIGURES.....	vi
LIST OF TABLES.....	vii
ABSTRACT.....	01
CHAPTER 01: INTRODUCTION	
1.1 Motivations.....	02
1.2 Contribution Summary.....	03
1.3 Thesis Orientation.....	04
CHAPTER 02: LITERATURE REVIEW	
2.1 Skeleton/Human Body Detection.....	05
2.2 Fixed Position Detection.....	06
2.3 Computation of Rotating Position.....	06
2.4 Gesture Control.....	07
CHAPTER 03: SYSTEM DESCRIPTION	
3.1 Microsoft Kinect	08
3.1.1 Introduction to Kinect Sensor.....	08
3.1.2 Kinect SDK.....	09
3.1.3 Kinect Hardware.....	10
3.1.4 Kinect Drivers.....	10
3.2 Microsoft Visual Studio (IDE).....	10
3.3 Display/Screen.....	11
CHAPTER 04: SYSTEM DESIGN	
4.1 System Implementation.....	12
4.1.1 System Implementation flow.....	12
A. User and Kinect.....	12
B. Position and Screen Setup.....	13

C. Screen.....	13
D. Position Comparison.....	13
E. Input.....	13
F. Computation.....	13
G. Compare and Coordinate.....	13
H. Display.....	13
4.2 Process flow	14
4.3 Application flow	15

CHAPTER 05: SYSTEM IMPLEMENTATION

5.1 Color Stream	16
5.2 Tracking (Skeleton).....	17
5.3 Gesture Recognition	21
5.4 User Interface	22
5.5 Computation of Two Points.....	23

CHAPTER 06: EXPERIMENTAL EVALUATION

6.1 Testing.....	25
6.2 Checking Values.....	28
6.3 Performance.....	29

CHAPTER 07: CONCLUSION AND FUTURE WORKS

7.1 Conclusion.....	31
7.2 Future work.....	31

REFERENCES.....	33
------------------------	-----------

LIST OF FIGURES

Figure 01	Kinect components.....	08
Figure 02	Hardware and software interaction with an application	09
Figure 03	System implementation flow	12
Figure 04	System process flow.....	14
Figure 05	Application Flow of Proposed Model	15
Figure 06	User detection by Kinect sensor	17
Figure 07	(a) Skeleton joints found by Microsoft Kinect.....	18
	(b) Joint structure on a user.....	18
Figure 08	Illustration of the skeleton space	19
Figure 09	(a) Body joints (Standing and Seated Mode)	20
	(b) Body joint points in seated mode.....	20
Figure 10	Minimum distance for tracking	21
Figure 11	User Interface of our system.....	23
Figure 12	(a) Final system testing.....	27
	(b) Before Tracking State.....	27
	(c) After Tracking State.....	28
Figure 13	Kinect sensor and user.....	30

LIST OF TABLES

Table 01	Color formats and short description	16
Table 02	Dataset	25
Table 03	Accuracy measurement of Kinect sensor.....	29

ABSTRACT

Now a days shopping became a daily essential part of our regular life. When we analysis in both online and going physically in the market we get to come up in a point that everyday people are remaining busy with the shopping in any kind of purpose in every day. We observed that there are lots of markets in Dhaka city but we felt strange that the markets are being busy with the huge crowd of people especially for the women. Besides, we observed that people often faces trial room problem in most of the market because of a long number of queue which is very disappointing for the customers. Moreover, the blessing of technology creates a nice and easy environment to the customers with lot of opportunities on shopping. So, to enhance users shopping experience and to spend less time on maintain queue for fitting room, our goal is to presents a virtual mirror model using gesture recognition technique. This allows a person to check how a dress looks like and which color is suitable on a person's body. Moreover, it shows user's body measurement when users try on virtual clothes. In the proposed model, we used Microsoft Kinect sensors to track user skeleton movement and depth image. The cloths are simulated in such a way that will represent an environment for the user like the mirror.

CHAPTER 01

INTRODUCTION

1.1 Motivations

In this modern era of revolution everyone is depending more and more on technology. According to this flow of development in technology the common definition of shopping is also changing by time to time. Now there comes online shopping which got popularity. Now-a-days online shopping or shopping through web is getting more popular because it is saving huge amount of valuable time of the shoppers and also reducing other hassles. Moreover, online shopping is being accepted widely all over the world. More than 85% of world's population has ordered goods over the internet during the recent years [1]. People are getting more attracted to the online shopping because of its extra features or offers like free home delivery, cash on delivery, and different kinds of discounts. However, it has a significant drawback- this method is not being accepted by all peoples as there is no surety that the delivered goods or cloths will be according to the expectation of the customer. Although customers can find all the description of the cloth like style, size, color fabric and other features through the web page, but they cannot determine whether the cloth is exactly suitable for their own style, color, size and other aspects. Therefore, the delivered clothes might also not fit the customers [2]. Previously, a number of researchers worked on this area to overcome the problems of online shopping. The researchers came up with an idea of virtually try the dresses or clothes so that the user do not have to try it physically [3, 4]. So, to enhance users shopping experience and to spend less time on maintain queue for fitting room, our goal is to presents a virtual mirror model using gesture recognition technique. This allows a person to check how a dress looks like and which color is suitable on a person's body. Moreover, it shows user's body measurement

when users try on virtual clothes. In the proposed model, we used Microsoft Kinect sensors to track user skeleton movement and depth image. The cloths are simulated in such a way that will represent an environment for the user like the mirror. In addition, we developed an algorithm for matching up all the motions between the virtual cloths and the human body.

1.2 Contribution Summary

We have read some research papers. Among them we have chosen “Magic Mirror Using Kinect,” by A. B. Habib, A. Asad, W.B. Omar, BRAC University (2015).

By studying this research paper, we found the following limitations:

- User need to move to adjust the cloth within his or her body.
- Dresses were not accurate to the body shape.
- Used no user-interface.
- Used fixed number of static dress

To overcome the limitations, we proposed a concept of real time virtual dressing room [3]. As mirrors are indispensable objects in our lives, the capability of simulating a mirror on a computer display or screen, augmented with virtual scenes and objects, opens the door for solving the major drawback in online shopping concept. An interactive Mirror could enable the shoppers to virtually try clothes, dresses using gesture-based interaction [5]. Therefore, we have proposed a method having the following features.

- Use a static position.
- Display size of dresses.
- Use hand gesture and improved user interface.
- Dynamic dresses based on user size

In this thesis, gesture based interaction techniques are used in order to create a virtual mirror for the virtualization of various clothes. When trying a new cloth the shoppers look into the shop's mirror in a shop. We create the same impression but for virtual clothes that the customer can choose independently. For that purpose, we replace the real mirror by a display that shows the mirrored input of a camera capturing the body skeleton of a person.

1.3 Thesis Orientation

In this thesis we will discuss about

- Chapter 02- In this part we will discuss about papers which we read to complete this thesis.
 - Chapter 03- The machineries or software environment or tools we have used to do the thesis is described in this part.
 - Chapter 04- How we design our system is described in this part with diagrams.
 - Chapter 05- Here we talk about how our system is working after implementing the design.
 - Chapter 06- After implementing the system, how it works, the results and performance of our system is described in this part.
- And lastly,
- Chapter 07- Conclusion and Future Work are given in this part.

CHAPTER 02

LITERATURE REVIEW

2.1 Skeleton/Human Body Detection

To detect the human body first we needed to detect the skeleton. To achieve this task there were many researches done previously. Before Kinect was developed researchers used different method to detect the human structure. In [6], Lee *et al.* worked with various algorithms and process to get the body structure to identify the human body by edge-based body part identification. Using the feature of the edge then the body part was identified and tracked in 2D environment. In [7], Gavrilu *et al.* went for a multiple view approach. They used multiple cameras to capture simultaneous image sequence of a subject. Then they compared with a HIA database to identify and track the human body movement. Du *et al.* proposed a model considering depth image data in [8]. They did the RGB-D mapping where the using joint algorithm and shape-based alignment. These were the base of the development of skeleton tracking. Then the researcher used these researches to compile a unique system with the sensors and algorithm named Kinect. In [9], Vera *et al.* created an augmented reality based system using Kinect to build a virtual augmented mirror. Where there would be character who would talk with the audience. Here they used the Kinect control algorithm and depth data to make the avatar move and interact. Aswini Vijay *et al.* worked on face detection to identify the orientation of the subject which enabled them to identify a specific region to implement the algorithm to identify body structure properly [10]. Thought Kinect has the skeleton detection algorithm but if the subject is facing backward then system will not work efficiently. This paper used the face detection to be the first system flow to identify the skeleton orientation and the skeleton of the subject.

2.2 Fixed Position Detection

To make the system more efficient and more functional we needed to fix the position of the user. A fixed position will enable us to get the skeleton data more accurately and to show user the measurement of a cloth. In [5], Presly *et al.* created a virtual fitting room based on depth data of the Kinect. They used unity environment and to make mesh to create an avatar which would be generated by user's body structure. In that system the body joint coordinate detection process is implemented and before starting the body tracking process the user must be in a fixed position and gesture to initiate the calibration process.

2.3 Computation of Rotating Position

The rotation of a user in front of the Kinect needed to be detected to make the cloth fit in the right place. If the rotational angle is changed the cloth simulation need to be changed also. For cloth generation it is divided into particle which is a vector dimension. In [12], Billy *et al.* worked with Kinect to develop a system and a user-friendly application to build an augmented fitting room. They used NUI Library and to make it more efficient they emphasized of the cloth simulation. In the cloth simulation they divided the cloth into particle and each particle represented a vector dimension which could be calculated and refined using different differential algorithm. Ordinary differential equation is used to measure the velocity and the force of each particle in the system's each image frame which eventually creates a proper cloth simulation. In these equations the most less error range is generated by Runge-Kutta equation.

2.4 Gesture Control

To make the system more user friendly the system need to have an interaction point where it can interact with the user. To make that happen a gesture control was needed which will be used to change simulated cloth in the image canvas. To achieve that, Ziquan *et al.* built a system to try cloth in virtual screen, which would help in future in shopping experience [13]. Their system also has gesture control to select dress and to screenshot capturing functions so that user can take a photo and share with other via QR code. They built the system using depth data of the Kinect. In this system the gesture was emphasized as they fail to make a 3D based model cloth. They created a proper interface so that the user can easily understand what is going on in the system and they can control the cloth simulation.

CHAPTER 03

SYSTEM DESCRIPTION

3.1 Microsoft Kinect

3.1.1 Introduction to Kinect Sensor

Kinect is a motion sensing input device developed by Microsoft in early 2010 [11]. Kinect is mainly used in Xbox 360 console and for Windows PCs [11]. For Windows, a Kinect sensor consists of an RGB camera which can store up to three channel data of resolution 1280×960 . It has an IR (infrared) emitter which emits lights beams and an IR depth sensor that reads the reflected beams and process the information to measure the distance between the object and the sensor. It also has multi-array microphone that can capture sound and detect the location of the source and direction of the audio wave. It has a practical ranging limit of 40cm – 3.5m for Windows and the frame rate is 30 FPS (Frames per Second) [14]. Figure 1 depicted a picture of a Kinect.

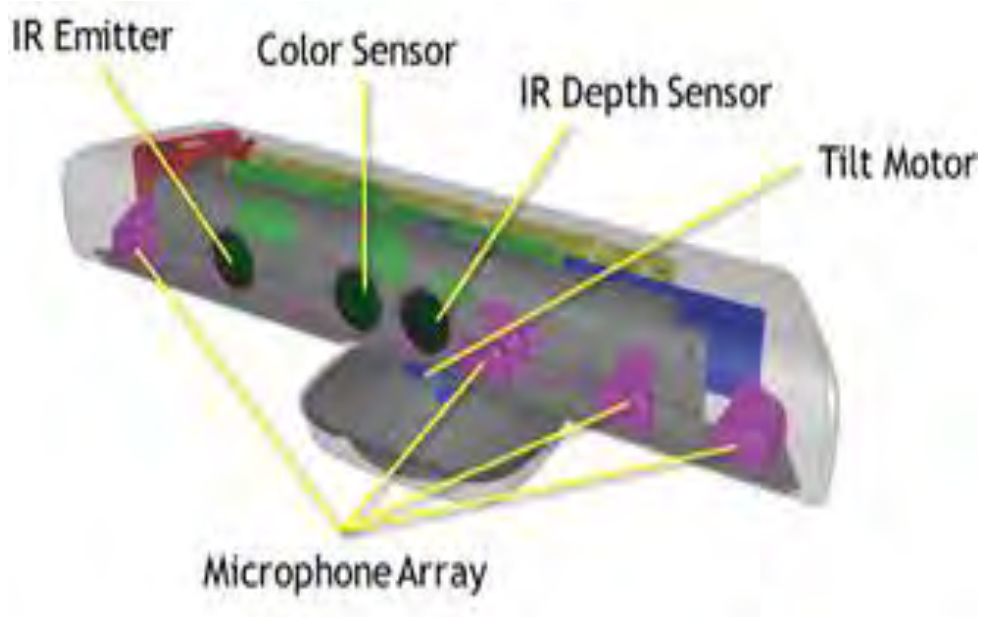


Figure 1: Kinect components [14].

3.1.2 Kinect SDK

Kinect SDK is developed to enable developers to develop applications in C++, C# or Visual Basic by using Microsoft Visual Studio [4]. It is capable of capturing front body 2D motion, gesture, facial and voice recognition [11], skeletal tracking and advanced audio capabilities [15]. To setup virtual mirror we need Kinect sensor to record skeleton and depth data and capture the RGB video stream.

Core of the Kinect for the set of Windows API, supports fundamental image and device management features like access to the Kinect sensors that are connected to the computer, access to image and depth data streams from the Kinect image sensors and delivery of a processed version of image and depth data to support skeletal tracking. Figure 2 shows interaction method of sensor Hardware with software of an application.

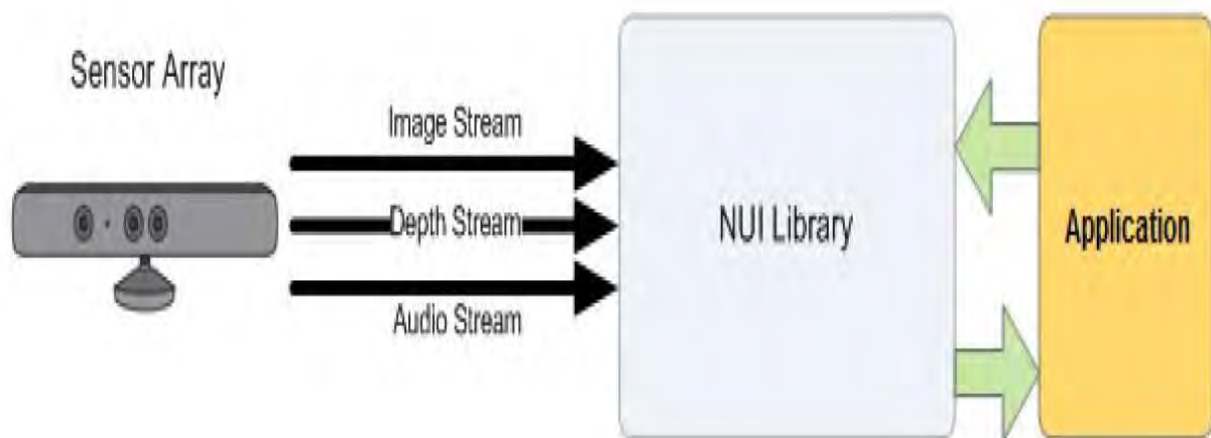


Figure 2: Hardware and software interaction with an application [3].

In addition, the software has the capability to recognize and track human body. The software runtime converts depth data into about 20 skeleton joint points of human body to track up to two persons in front of the camera [16].

3.1.3 Kinect Hardware

Including the Kinect sensor and the USB hub, through which the sensor is connected to the computer [3].

3.1.4 Kinect Drivers

Microsoft provides Windows (7, 8, 8.1, or 10) drivers for the Kinect sensor [3].

3.2 Microsoft Visual Studio (IDE)

Microsoft visual studio is an integrated development environment (IDE) from Microsoft. It is an integrated solution which enables the users to develop console and graphical user interface applications along with Windows Forms Applications (WFP), web sites, web applications, web services etc [17]. Visual studio supports almost all kinds of programming languages including built-in languages such as C, C++ [18] (via Visual C++), VB.NET (via Visual Basic .NET), C# (via Visual C#), and F# (as of Visual Studio 2010 [19]). All of these languages are built on top of the .NET Runtime (known as Common Language Runtime or CLR) and produce the same intermediate output in Microsoft Intermediate Language (MSIL) [10]. Like any other IDE, it includes a code editor that supports syntax highlighting and code completion using IntelliSense for variables, functions, methods, loops and LINQ queries [21]. IntelliSense is supported for the included languages, as well as for XML and for Cascading Style Sheets and JavaScript when developing websites and web applications [22, 23]. Autocomplete suggestions appear in a modeless list box over the code editor window, in proximity of the editing cursor. In Visual Studio 2008 onwards, it can be made temporarily semi-transparent to see the code obstructed by it [21]. The code editor is used for all supported languages.

3.3 Display/Screen

To give the user a mirrored image impression there is a screen or display in front of the user. Data processing has to be accurate and mirroring the real time video essential in order to give the user an impression of mirror.

CHAPTER 04

SYSTEM DESIGN

4.1 System Implementation

Figure 3 shows a detailed block diagram of system implementation of our proposed model.

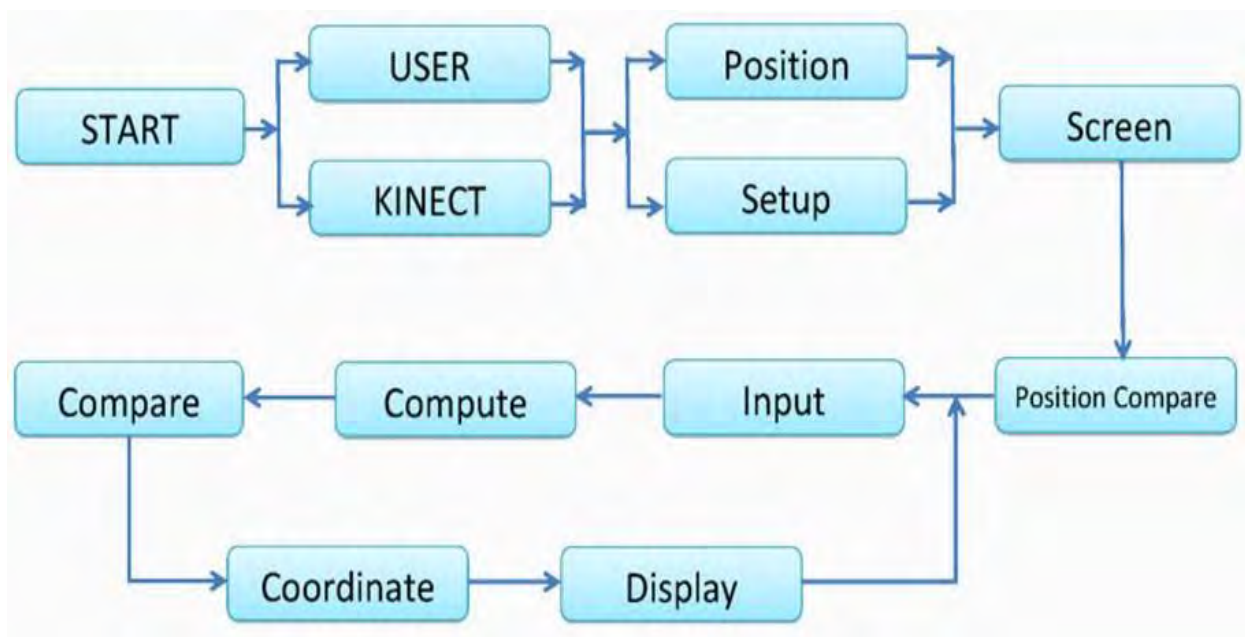


Figure 3: System implementation flow.

4.1.1 System Implementation flow

A. User and Kinect

Human body act as input source and by Kinect we take that input. Both Kinect and raw data from user are taken parallel. After that all information passes to the next session as input.

B. Position and Screen setup

After taking raw values from user and Kinect the system measures the position of user and set screen setup.

C. Screen

Screen will process the raw data and change it some frame model and will show into the screen.

D. Position Comparison

In this stage our code will compare each array with previous array of frame to identify the rotation of human.

E. Input

At this stage system will save all the raw data for further calculation.

F. Computation

The raw data are compiled and also compute the dress model data.

G. Compare and Coordinate

From previous step's raw values, it will compare human model values with dress model values take some point to coordinate them.

H. Display

After doing all this process our system will display the final output. Then it will prepare itself for next input.

4.2 Process flow

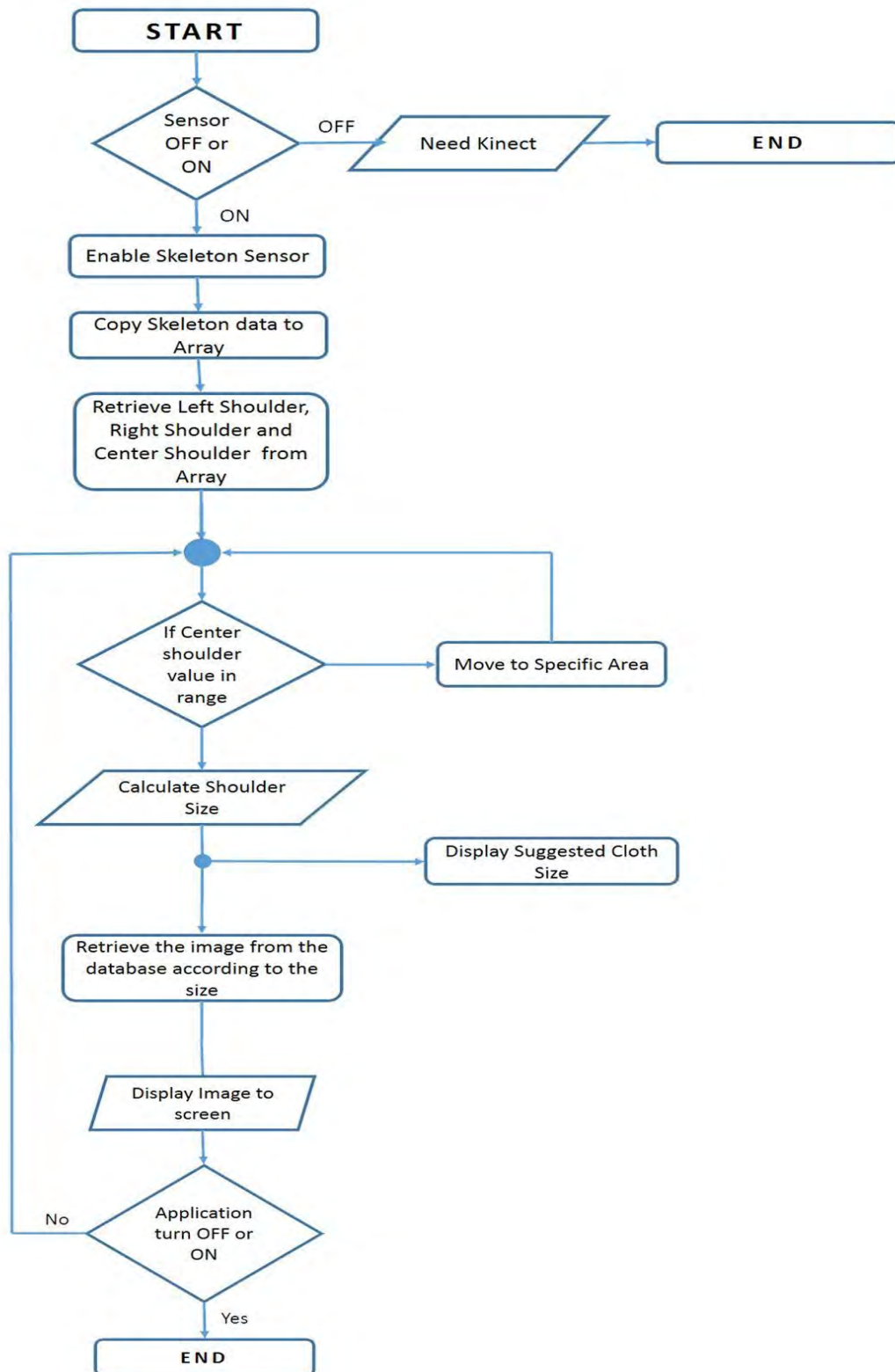


Figure 4: System process flow

4.3 Application flow

Figure 5 shows the application flow of the proposed model.

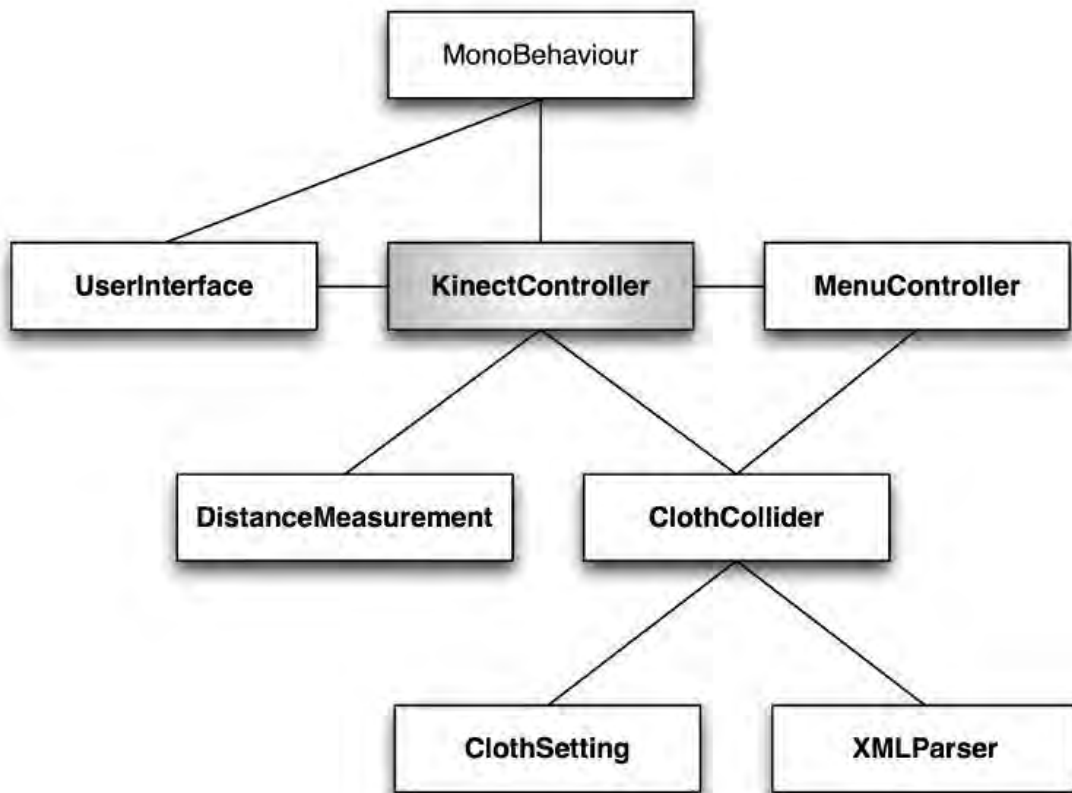


Figure 5: Application Flow of Proposed Model.

Our application flow starts from Kinect Controller. It works as a main class which also controls other sub class like measurement, cloth collider. There is also a user interface, a Menu Controller which is derived from Mono-behavior.

CHAPTER 05

SYSTEM IMPLEMENTATION

5.1 Color Stream

Color Stream is mainly used to display the view of the camera from the Kinect to the display or screen. Kinect provides color image data in different formats and resolutions. The color stream is displayed in an image on WPF which is continuously updated by each color stream packet.

Table 1: Color formats and short description [24].

Color Format	Description
RGB	32-bit, linear X8R8G8B8-formatted color bitmaps RgbResolution640x480Fps30 or RgbResolution1280x960Fps12 or YuvResolution640x480Fps15 color formats
YUV	16-bit, gamma-corrected linear UYVY-formatted color bitmaps RawYuvResolution640x480Fps15 color data formats YUV data is available only at the 640x480 resolution and only at 15 fps.
Bayer	32-bit, linear X8R8G8B8-formatted color bitmaps RawBayerResolution1280x960Fps12 or RawBayerResolution640x480Fps30 color formats

The format defines whether the data stream is encoded as RGB, YUV, and Bayer. Resolution is about how the bandwidth is used. Through USB connection Kinect sensor provides a range of bandwidth for passing data. High-

resolution images send more data per frame and update less frequently, while lower-resolution images update more frequently, with some loss in image quality due to compression [11]. Table 1 specifies available formats for color stream data [24].

To display the sensor camera view to screen we have used Color Frame class. All the image data and format are present in this class. We have used RgbResolution640x480Fps30 as the color format to achieve maximum display output. The image data is stored as a pixel array. Then a Writeable Bitmap is declared to store the pixel data and displayed through display source.

5.2 Tracking (Skeleton)

One of the best features of Microsoft Kinect for Windows SDK is Skeletal Tracking. By using the Infrared (IR) camera, the sensor can track people and follow body actions. The Skeletal Tracking method can recognize up to six persons present in the field view. Out of six persons the method can track up to two people in detail. Figure 6 shows the joint points of a human body detected by Kinect [17].

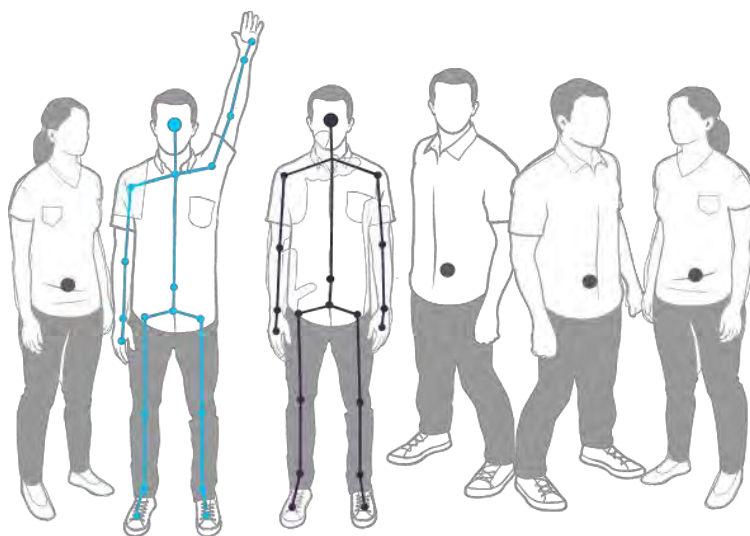


Figure 6: User detection by Kinect sensor [7].

The method can recognize or locate 20 joint points as skeleton of the person. These 20 joint points are the locations of different parts of the tracked person that in total make a human skeleton. Figure 7 shows the joint points of a human body detected by Kinect.

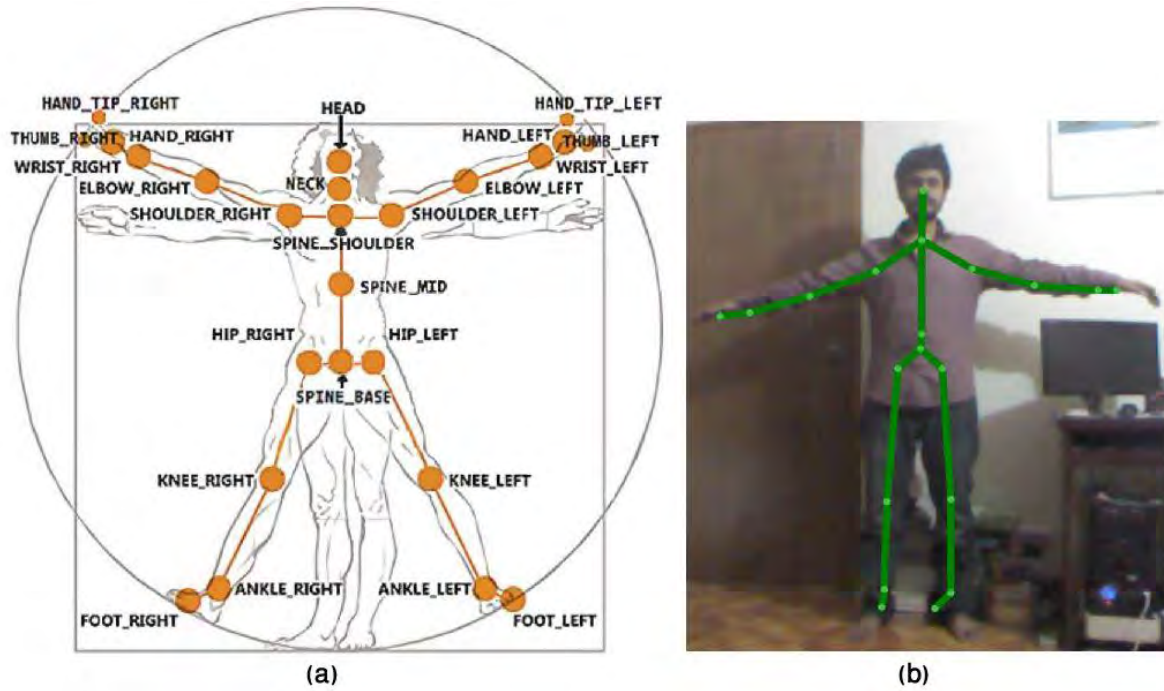


Figure 7: (a) Skeleton joints found by Microsoft Kinect [3], (b) Joint structure on a user.

To track human body joint point's accurately Skeleton data is stored 3D position data for human skeleton. The data of each skeleton joint are stored as X, Y, and Z coordinates. These coordinates are measured in meters by default. Figure 8 shows the x, y, and z axis of the sensor.

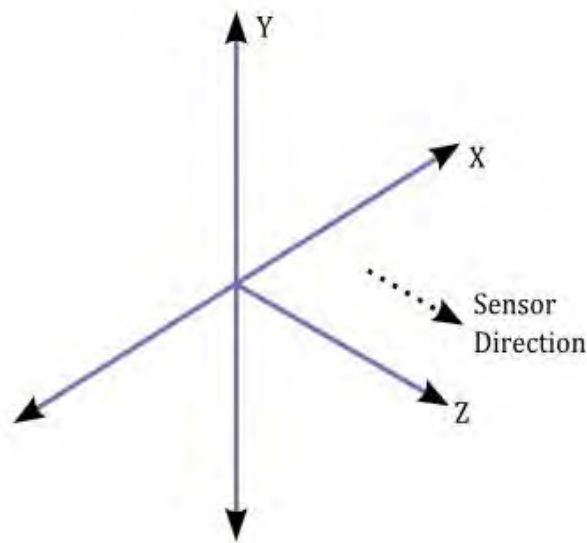


Figure 8: Illustration of the skeleton space [14].

The above figure is a right-handed coordinate system in which the Kinect sensor is placed at the origin. The center point of the sensor is the origin of the coordinate meaning that the coordinate $(0, 0, 0)$ lies at the middle of the sensor.

X-axis: The positive x-axis increases on the left side.

Y-axis: Positive y-axis is increased in upward and decreased in downward.

Z-axis: Positive z-axis is increased in the direction of Kinect sensor.

These coordinates can be determined by the following method:

JointType.Position.X

JointType.Position.Y

JointType.Position.Z [17]

The skeletal tracking has two different modes of tracking user. One is standing mode and other one is seated mode. In standing mode, the tracking algorithm can track all 20 joint positions of the user. Whereas the seated mode can track the 10 joint points of the upper body starting from head to Hip Center.

Standing is the default mode for the tracking algorithm. Figure 9 shows the tracking joint points in standing and seated mode.

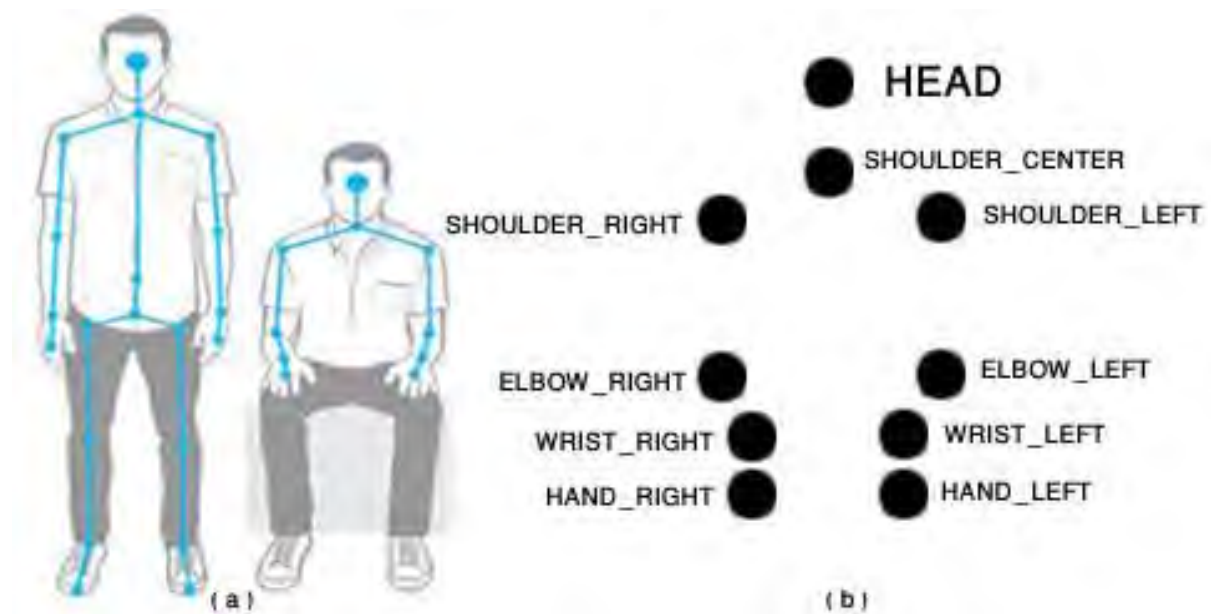


Figure 9: (a) Body joints (Standing and Seated Mode), (b) Body joint points in seated mode.

To successfully track and get the values the user must have to maintain a minimum distance from Kinect sensor. The minimum distance is 2.1 meter or 6.89 feet. Figure 10 indicates that the user would have to be at a minimum distance from the Kinect to be able to successfully track by Kinect [1].

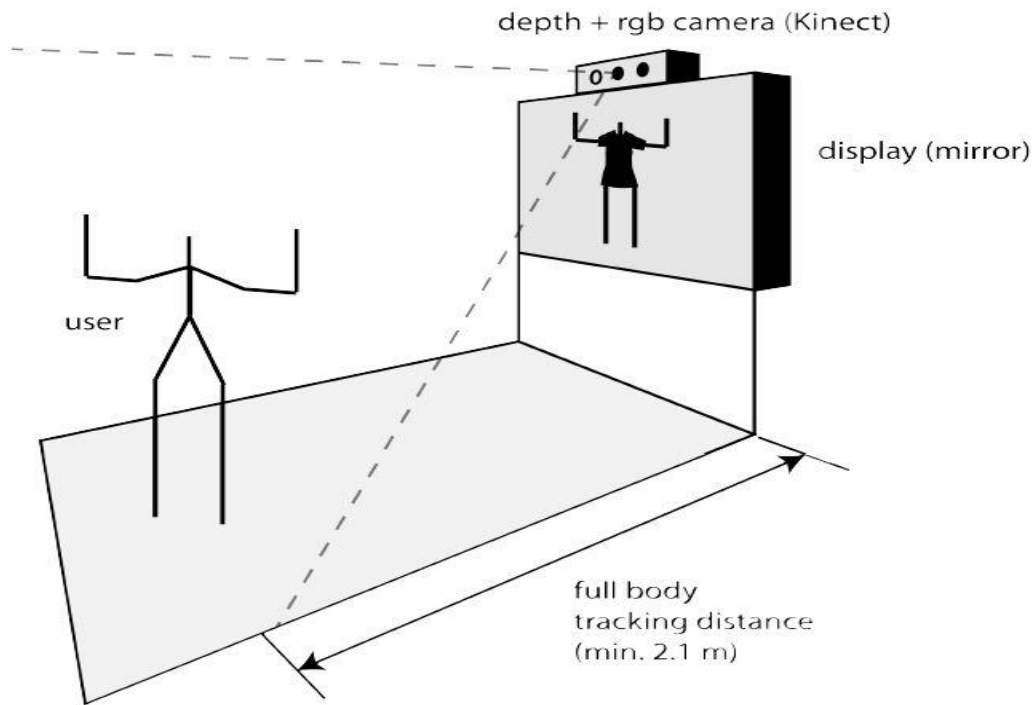


Figure 10: Minimum distance for tracking [1].

As our one of the goal is to view the user size of the dress, we have used skeleton coordinates to calculate body measurement of the user. We have implemented an algorithm that measures the body width by taking the values of these skeleton coordinates. The algorithm calculates the body measurement by the coordinates of shoulderLeft, shoulderRight and shoulderCenter joint points. shoulderCenter, hipCenter are used to calculate the position of the user.

5.3 Gesture Recognition

In our final system we have used Gesture Recognition technology in the user interface. This gesture recognition enables the user to change the clothes the by swipe of their hand. It is very efficient as the user stands in a little distance from display or screen and cannot use touch-screen or other technology to change the cloths.

To gesture recognition work successfully we have used a built-in class called Recognizer. This class recognizes the left Swipe and right Swipe when the user hand is swiped over the Kinect sensor. When right hand swipe is detected then next cloth is shown in screen. And when left hand swipe is detected then previous cloth is shown in screen.

5.4 User Interface

As this is a virtual mirror and it can be implemented anywhere from commercial cloth store to personal dressing mirror, there is no need of external display for the application interface. The virtual mirror is used to show the interface and it is controlled by gesture recognition feature of Kinect. Particularly swipe gesture will be implemented and will be distinguished by right and left swipe method. This swipe method can be tracked and implemented by using the Kinect sensor. By using this feature user will be able to choose their desired clothes to display. Figure 11 shows the xml view of our user interface.

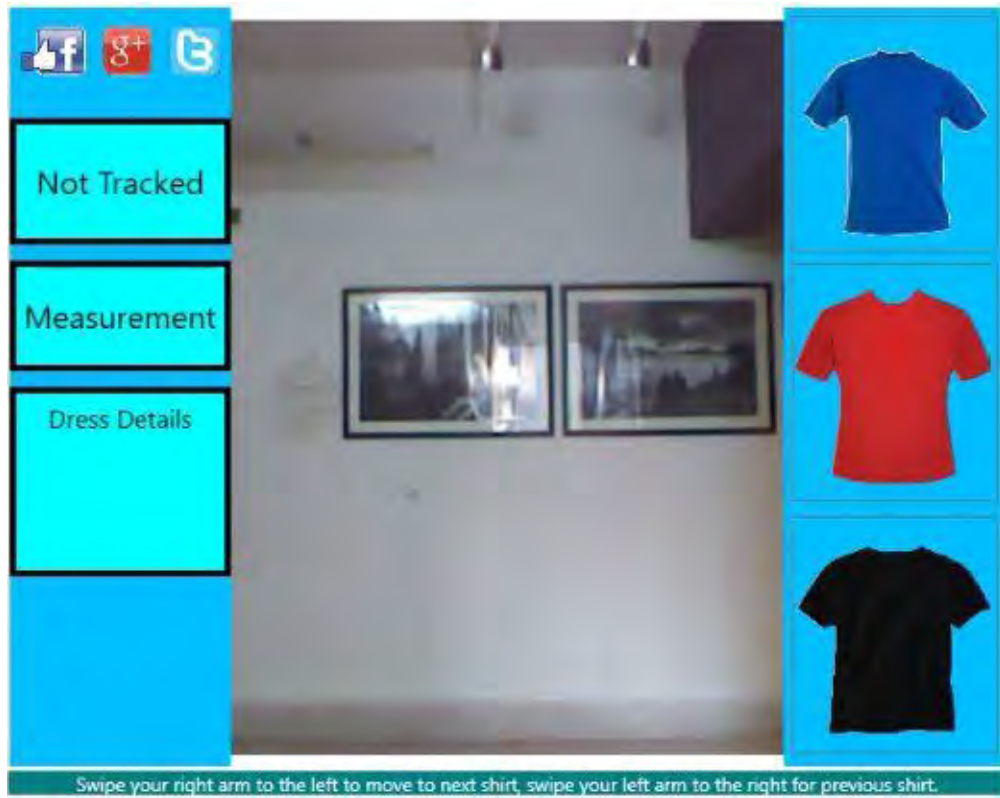


Figure 11: User Interface of our system.

5.5 Computation of Two Points

As stated earlier one of the main purposes of our system is to display preferred size of user clothing. For this computation we have used Kinect's Skeleton Position method. We have created an algorithm that calculates the measurement between two shoulders by taking the values of left Shoulder and right Shoulder. Kinect sensor maps the skeleton values in 3D coordinate (X, Y, Z). We have mapped the coordinate in 2D by fixing the Z coordinate. To calculate the value, we have used the basic method of 2D system:

$$W = \frac{Y_2 - Y_1}{X_2 - X_1} \dots\dots\dots (1)$$

$$W_R = \frac{Y_{LS} - Y_{CS}}{X_{LS} - X_{CS}} \dots\dots\dots (2)$$

$$W_L = \frac{Y_{LS}-Y_{CS}}{X_{LS}-X_{CS}} \dots\dots\dots (3)$$

Here,

W = Distance of two coordinate points

W_L = Distance of left shoulder coordinate point

W_R = Distance of right shoulder coordinate point

Y_2 = next value of y-axis and Y_1 = previous value of y-axis

X_2 = next value of x-axis and X_1 = previous value of x-axis

Y_{LS} = left shoulder y-axis value and Y_{RS} = right shoulder y-axis value

X_{LS} = left shoulder x-axis value and X_{RS} = right shoulder x-axis value

Y_{CS} = center shoulder y-axis value and X_{CS} = center shoulder x-axis value

Using the above equation, we have calculated the measurement of right shoulder by calculating the coordinate distance between right Shoulder and center Shoulder. Then the measurement of left shoulder has been determined by calculating the coordinate distance between left Shoulder and center Shoulder.

CHAPTER 06

EXPERIMENTAL EVALUATION

6.1 Testing

During the testing phase main attention was given to all the implemented functions and how the data was presented on different users. To evaluate the performance of the system, we have taken 20 test subjects (Users) and measured their body width. Table 2 holds the 20 subjects body measurement dataset.

Table 2: Dataset.

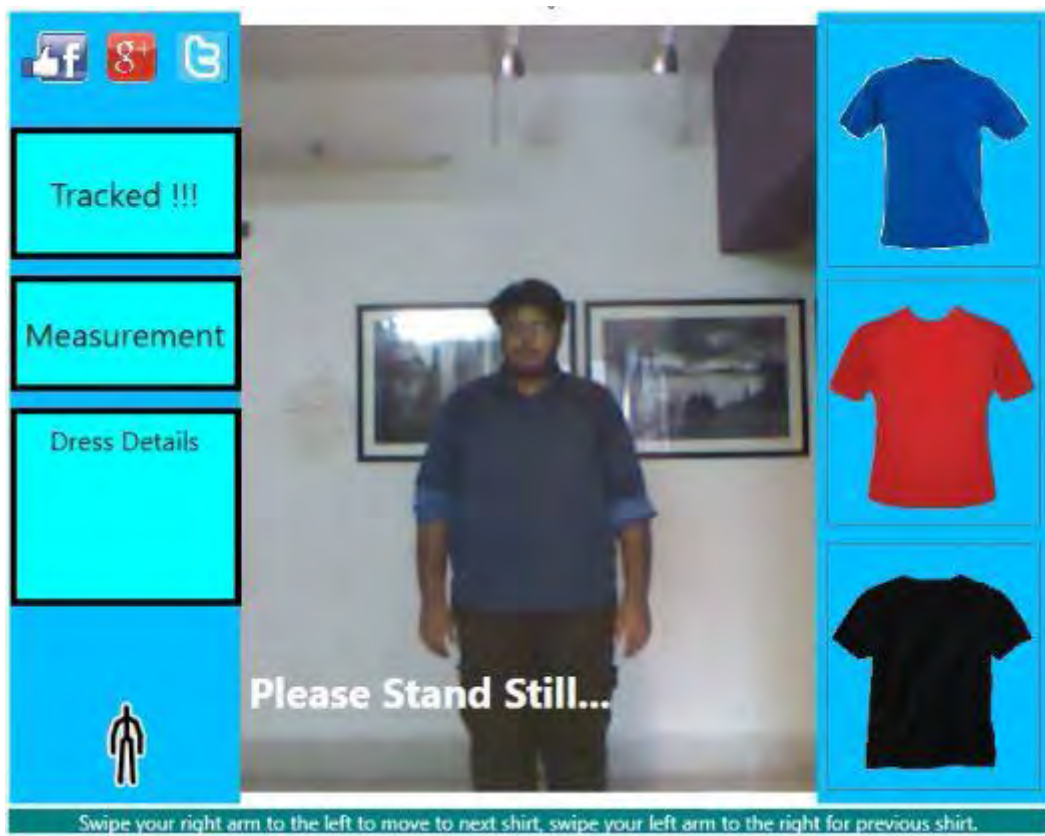
Sl.	Name	Gender	Height	Kinect Value	Preferred Size
1	Nafis	Male	Medium	694	Large
2	Nahyan	Male	Medium	604	Medium
3	Monir	Male	Long	614	Medium
4	Nargis	Female	Medium	620	Semi-Large
5	Mahmudul Hoq	Male	Short	563	Small
6	Jaber	Male	Short	579	Medium Small
7	Arifin	Male	Short	589	Medium
8	Shuvo	Male	Semi-Short	594	Medium
9	Peash	Male	Medium	572	Small
10	Bijoy	Male	Medium	621	Medium
11	Arpita	Female	Short	562	Medium-large
12	Oni	Male	Medium	616	Medium
13	Oporna	Female	Short	561	-

14	Mursida	Female	Short	553	-
15	Nivrito	Male	Short	611	Medium
16	Hirok	Male	Short	642	Large
17	Adnan	Male	Short	669	Large
18	Arko	Male	Long	618	Medium
19	Mahabub	Male	Long	578	Medium
20	Shoib	Male	Medium	624	Large

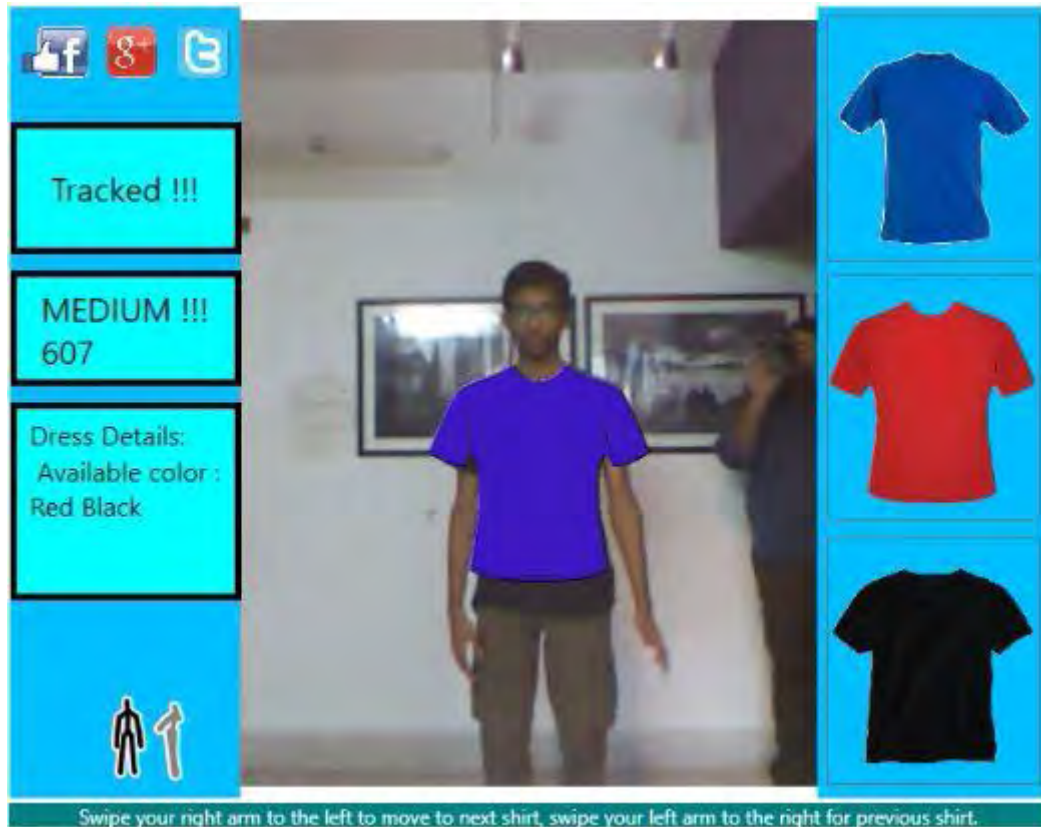
For testing purpose, we have instructed each of our subjects to stand in front of the Kinect sensor. After detecting the body measurement, we have noted the measurement to an excel sheet. After that we also noted their personal information such as name, gender and height. We have also asked about their preferred size when they go to shopping. We also noted their preferred size of clothes they wear. Moreover, we tested that our user static position is accurately working or not on each of the subject. Figure 12 shows the process of testing the system.



(a)



(b)



(c)

Figure 12: (a) Final system testing, (b) Before tracking State, (c) After Tracking State.

6.2 Checking Values

After testing phase, we categorized all the values in three categories. The categories are:

1. Large
2. Medium
3. Small

Based on these categories our system shows different size of garments. These sizes of garments are mainly dependent on system body measurement and the above mentioned categories.

6.3 Performance

To ensure the performance of our algorithm, we compared height calculated from KSDK (Kinect SDK) in terms of subject's original height. To this end, we recorded 30 frames (1s) of skeleton data from five subjects holding the standard standing position (T-pose) from a fixed distance to the Kinect sensor. The Kinect was placed 118 cm above the ground. Subject including both males and females were wearing T-shirt, jeans, and casual sneakers. We then calculated the standard deviation of each joint position for all the visible joints. Table 3 illustrates the accuracy of Kinect sensor measurement.

Table 3: Accuracy measurement of Kinect sensor.

Shoulder center-to-feet height (cm)	Shoulder center-to-feet height (cm) KSDK	Accuracy (%)
163.4	160.2	98.04
181.5	178.4	98.29
176.2	170.5	96.76
172.3	170.8	99.13
165.5	161.4	97.52

Figure 13 shows Kinect sensor’s perspective of a user [4].

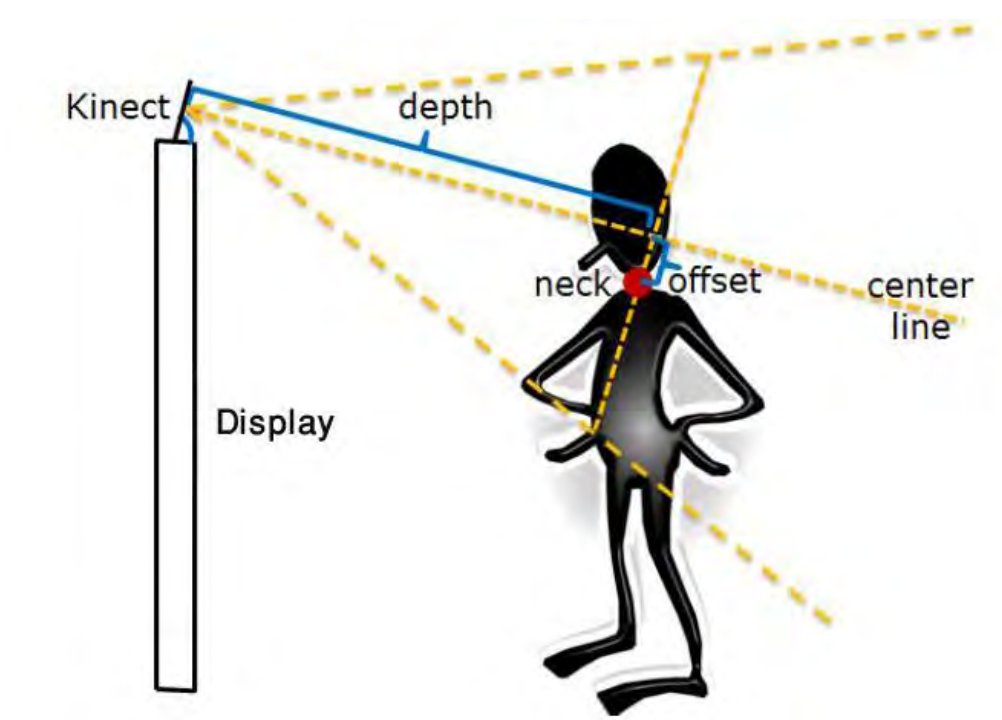


Figure 13: Kinect sensor and user [4].

The results are shown in the Table above. The first column of the Table lists our manual measurement of the vertical distance between the floors to the mid-point of the shoulder. This is the shoulder height that our clothes-body fitting algorithm expects to cover the virtual clothes.

CHAPTER 07

CONCLUSION AND FUTURE WORKS

7.1 Conclusion

In this paper, we introduce a virtual dressing room application where avatar and cloth generation, real time tracking technologies up to an overview of comparable virtual try-ons. Subsequently a closer look on the technologies and frameworks that were used for the implementation of the virtual dressing room was taken. After this the different aspects of the design process up to the construction of the garment models was highlighted. This is followed by the implementation, describing the cloth colliders and the behavior of the garment, for instance. In the last section the tests were executed, also discussing the output, the appearance and the interaction with the virtual dressing room. Overall, the presented virtual dressing room seems to be a good solution for a quick, easy and accurate try-on of garment. The Microsoft Kinect offers the optimal technology for a successful implementation. Compared to other technologies like augmented reality markers or real-time motion capturing techniques no expensive configurations and time-consuming build-ups are required. From this point of view, it is an optimal addition for a cloth store. A simple setup of the system can also be assembled at home since the minimum requirements are a computer with a screen and a Kinect.

7.2 Future work

- ✓ We have a plan to make more efficient virtual mirror that will give us the 3D output which will be more attractive and efficient. Actually we want make more efficient 3D modeling virtual mirror.
- ✓ Moreover, as we can say that virtual mirror is related as the alternative of

a dressing room, so we have a future plan to make a better use of the virtual mirror in the market as well.

- ✓ Besides, we have an interest to take the initiative to use the virtual mirror in the holography as well as in internet based application.

REFERENCES

- [1] U. Cheema, M. Rizwan, R. Jalal, F. Durrani, N. Sohail, “The Trend of online shopping in 21st century: Impact of enjoyment in tam model” Asian Journal of Empirical Research, vol. 3, no. 2, pp. 131-141.
- [2] L. Zhao, J. Zhou, “Analysis on the Advantages and Disadvantages of clothing Networking Marketing” International Journal of Business and social science, vol. 6, no. 4(1), pp.147-151, (2015).
- [3] A.B. Habib, A. Asad, W.B. Omar, “Magic Mirror Using Kinect,” BRAC University (2015).
- [4] S. Giovanmi, Y.C. Choi, J. Huang, E.T. Khoo, K. Yin, “Virtual try-on using Kinect and HD Camera,” MIG-2012, vol. 7660, pp. 55-65, (2012).
- [5] P. Presle, “A Virtual Dressing Room based on Depth Data,” Vienna Uninversity of Technology, Klosterneuburg, pp. 25-36, (2012).
- [6] M. W. Lee & R. Nevatia (2007, February). *Body Part Detection for Human Pose Estimation and Tracking*. Proceedings of the IEEE Workshop on Motion and VideoComputing.
- [7] D. M. Gavrilu, L. S. Davis (1996, June). *3-D model-based tracking of humans in action: a multi-view approach*. Proceedings of the 1996 Conference on Computer Vision and Pattern Recognition, San Francisco.
- [8] H. Du, P. Henry, X. Ren, M. Cheng, D. B. Goldman, S. M. Seitz, & D. Fox, (2011, September). *RGB-D Mapping: Using Depth Cameras for Dense 3D Modeling of Indoor Environments*. Proceedings of the 13th international conference on Ubiquitous computing, Beijing, China.
- [9] L. Vera, J. Gimeno, I. Coma & M. Fernández, (2011, September). *Augmented Mirror: Interactive Augmented Reality based on Kinect*. Proceedings of the 13th IFIP TC 13 International Conference on Human-Computer Interaction, Lisbon, Portugal.

- [10] A. Vijayaraghavan, Induhumathi T.A, J. Chopra A.R.N, K. Miracline R, "A REAL Time Virtual Dressing Room Application Using OpenCV", Anna University: Chennai 600 025, April 2014.
- [11] [online] <http://www.dmi.unict.it/~battiato/CVision1112/Kinect.pdf>.
- [12] H.C.Y. Billy, M.C.K. Charles, N.K.K. Kit, W.Kenneth, A.Tam, "Intelligent Mirror for Augmented Fitting Room Using Kinect - Cloth Simulation", Department of Computer Science, University of Hong Kong.
- [13] L. Ziquan, S. Zhao, "Augmented Reality: Virtual fitting room using Kinect", Department of Computer Science, School of Computing, National University of Singapore, December, 2011.
- [14] [online] <https://msdn.microsoft.com/en-us/library/jj131033.aspx>.
- [15] H. Fairhead, "All About Kinect,". Retrieved from <http://www.i-programmer.info/babbages-bag/2003-kinect-the-technology-.html>.
- [16] G. Yolcu, S. Kazan, and C. Oz, "Real Time Virtual Mirror Using Kinect," Baikan journal of Electrical & Computer Engineering, vol. 2, no. 2, pp. 75-78, (2014).
- [17] [online] <http://home.hit.no/~hansha/documents/microsoft.net/tutorials/introduction%20to%20visual%20studio/Introduction%20to%20Visual%20Studio%20and%20CSharp.pdf>.
- [18] Brenner, Pat (19 July 2013). "C99 library support in Visual Studio 2013", Visual C++ Team Blog. Microsoft.
- [19] D. Chai and K. N. Ngan, Face Segmentation using Skin-Color Map in Videophone Applications, IEEE Transactions on Circuits and Systems for Video Technology, vol. 9, no. 4, pp. 551-559, (1999).
- [20] A. Sur, Visual Studio 2012 and .NET 4.5 Expert Development Cookbook, vol.1, Chapter No. 1 "Introduction to Visual Studio IDE Features," (2013).

- [21] Guthrie, Scott. “Nice VS 2008 Code Editing Improvements”, July 28, 2007.
- [22] Guthrie, Scott. “VS 2008 JavaScript IntelliSense,” June 22, 2007.
- [23] Guthrie, Scott. “VS 2008 Web Designer and CSS Support,” July 25, 2007.
- [24] [online] <https://msdn.microsoft.com/en-us/library/jj131027.aspx>.
- [25] M. Kotan, and C. Oz, “Virtual Mirror with Virtual Human using Kinect Sensor,” 2nd International Symposium on Innovative Technologies in Engineering and Science, Karabuk University, Turkey, pp. 730-738, (2014).
- [26] [online] <http://tutorial.math.lamar.edu/Classes/DE/EulersMethod.aspx>.
- [27] [online] <https://www.math.ksu.edu/math240/book/chap1/numerical.php>.
- [28] [online] http://www.myphysicslab.com/runge_kutta.html.